Notes on Geometric-Algebra Quantum-Like Algorithms

Diederik Aerts [1] and Marek Czachor [1,2]
[1] *Centrum Leo Apostel (CLEA) and Foundations of the Exact Sciences (FUND)*
Vrije Universiteit Brussel, 1050 Brussels, Belgium
[2] *Katedra Fizyki Teoretycznej i Informatyki Kwantowej*
Politechnika Gdańska, 80-952 Gdańsk, Poland

In these notes we present preliminary results on quantum-like algorithms where tensor product is replaced by geometric product. Such algorithms possess the essential properties typical of quantum computation (entanglement, parallelism) but employ additional algebraic structures typical of geometric algebra — structures absent in standard quantum computation. As a test we reformulate in Geometric Algebra terms the Deutsch-Jozsa problem.

I. INTRODUCTION

Quantum algorithms are based on tensor products. Common wisdom states that tensor products are needed for entanglement. A similar situation was encountered in the early 1990s in connectionist systems, and led to the paradigm known as tensor product representations [1]. However, nowadays the cognitive science community seems to depert from tensor product representations in favor of their "compressed forms" such as Binary Spatter Codes (BSC) [2] or Holographic Reduced Representations (HRRs) [3].

The main reason why BSC and HRRs replace tensor product representations is that tensor multiplication expands dimensions of the associated liner spaces (tensor product of two n-tuples is an n^2-tuple). HRRs, for example, replace tensor product with circular convolution, an operation that does not change the dimension (circular convolution of two n-tuples is again an n-tuple). Circular convolution is often referred to as a compressed form of the tensor product. Similarly, in BSC one replaces tensor products by XORs of binary strings. Quite recently, following the general program of investigating similarities and differences between AI, semantic analysis, and quantum information [4], we have reformulated BSC in terms of Geometric Algebra (GA) [5, 6, 7]. This reformulation was made possible by the observation that XOR has a natural representation at the level of geometric product.

In these notes we present a similar reformulation of the Deutsch-Jozsa algorithm [8]. As one can see, after minor modifications the GA algorithm works analogously to the quantum one. Accordingly, it is possible that GA algorithms can perform more general tasks until now reserved for quantum computation. The fact that it was easy to reformulate in a GA way the Deutsch-Jozsa problem is very encourageing.

Finally, let us mention that certain attempts of using GA for the purposes of quantum computation can be found in the literature [9, 10, 11]. Still, it seems that the approaches discussed so far reduce GA to the level of unitary operations or density matrices, i.e. objects that have a natural operator representation. In our approach, even "pure states" are represented by elements of GA. This is why we can perform operations on pure states that have no couterpart in standard quantum computation. In this sense our geometric algorithm may be regarded as something conceptually in-between quantum algorithms and HRRs or BSC.

II. ORIGINAL DEUTSCH-JOZSA ALGORITHM

We assume there exists an oracle performing

$$U_f|x\rangle|y\rangle = |x\rangle|y\oplus f(x)\rangle \tag{1}$$

where $f(x)\in\{0,1\}$. Now

$$U_f|x\rangle\Big(|0\rangle-|1\rangle\Big) = |x\rangle\Big(|0\oplus f(x)\rangle-|1\oplus f(x)\rangle\Big) \tag{2}$$

$$= |x\rangle\Big(|f(x)\rangle-|\neg f(x)\rangle\Big) \tag{3}$$

If $f(x)=0$ then

$$U_f|x\rangle\Big(|0\rangle-|1\rangle\Big) = |x\rangle\Big(|0\rangle-|1\rangle\Big) \tag{4}$$

If $f(x)=1$ then

$$U_f|x\rangle\Big(|0\rangle-|1\rangle\Big) = |x\rangle\Big(|1\rangle-|0\rangle\Big) = -|x\rangle\Big(|0\rangle-|1\rangle\Big) \tag{5}$$

The two cases imply

$$U_f|x\rangle\Big(|0\rangle - |1\rangle\Big) \;=\; (-1)^{f(x)}|x\rangle\Big(|0\rangle - |1\rangle\Big) \tag{6}$$

The Hadamard gate acts as follows

$$U_H|0\rangle \;=\; \frac{1}{\sqrt{2}}\Big(|0\rangle + |1\rangle\Big) \tag{7}$$

$$U_H|1\rangle \;=\; \frac{1}{\sqrt{2}}\Big(|0\rangle - |1\rangle\Big) \tag{8}$$

Let

$$U_{n+1} = \underbrace{U_H \otimes \cdots \otimes U_H}_{n+1} \tag{9}$$

Then

$$U_{n+1}|\underbrace{0\ldots0}_{n}1\rangle \;=\; \frac{1}{\sqrt{2^{n+1}}}\sum_{A_1\ldots A_n=0}^{1}|A_1\ldots A_n\rangle\Big(|0\rangle - |1\rangle\Big) \tag{10}$$

$$= \frac{1}{\sqrt{2^{n+1}}}\sum_{A_1\ldots A_{n+1}=0}^{1}(-1)^{A_{n+1}}|A_1\ldots A_n, A_{n+1}\rangle \tag{11}$$

$$= \frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle\Big(|0\rangle - |1\rangle\Big) \tag{12}$$

$$U_fU_{n+1}|\underbrace{0\ldots0}_{n}1\rangle \;=\; \frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle\Big(|0\rangle - |1\rangle\Big) \tag{13}$$

$$U_{n+1}U_fU_{n+1}|\underbrace{0\ldots0}_{n}1\rangle \;=\; \frac{1}{\sqrt{2^{n}}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}U_n|x\rangle|1\rangle \tag{14}$$

$$= \frac{1}{\sqrt{2^{n}}}\sum_{A_1\ldots A_n=0}^{1}(-1)^{f(A_1\ldots A_n)}U_n|A_1\ldots A_n\rangle|1\rangle \tag{15}$$

$$= \frac{1}{2^n}\sum_{A_1\ldots A_n=0}^{1}\sum_{B_1\ldots B_n=0}^{1}(-1)^{f(A_1\ldots A_n)}(-1)^{\sum_{k=1}^{n}A_kB_k}|B_1\ldots B_n\rangle|1\rangle \tag{16}$$

$$= \frac{1}{2^n}\sum_{B_1\ldots B_n=0}^{1}\Bigg(\sum_{A_1\ldots A_n=0}^{1}(-1)^{f(A_1\ldots A_n)}(-1)^{\sum_{k=1}^{n}A_kB_k}\Bigg)|B_1\ldots B_n\rangle|1\rangle \tag{17}$$

$$= \frac{1}{2^n}\sum_{(B_1\ldots B_n)\neq(0_1\ldots0_n)}\Bigg(\sum_{A_1\ldots A_n=0}^{1}(-1)^{f(A_1\ldots A_n)}(-1)^{\sum_{k=1}^{n}A_kB_k}\Bigg)|B_1\ldots B_n\rangle|1\rangle$$

$$+\frac{1}{2^n}\sum_{A_1\ldots A_n=0}^{1}(-1)^{f(A_1\ldots A_n)}|0_1\ldots0_n\rangle|1\rangle \tag{18}$$

$$= \cdots + \frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|0_1\ldots0_n\rangle|1\rangle \tag{19}$$

If $f(x) = f(0)$ for any x then

$$\text{RHS} \;=\; \cdots + \frac{1}{2^n}(-1)^{f(0)}\sum_{x=0}^{2^n-1}|0_1\ldots0_n\rangle|1\rangle = (-1)^{f(0)}|0_1\ldots0_n\rangle|1\rangle \tag{20}$$

If f is balanced then

$$\begin{aligned}
\text{RHS} &= \cdots + \frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|0_1\dots 0_n\rangle|1\rangle &(21)\\
&= \frac{1}{2^n}\sum_{(B_1\dots B_n)\neq(0_1\dots 0_n)}\Big(\sum_{A_1\dots A_n=0}^{1}(-1)^{f(A_1\dots A_n)}(-1)^{\sum_{k=1}^n A_kB_k}\Big)|B_1\dots B_n\rangle|1\rangle &(22)
\end{aligned}$$

It is sufficient to look at the $|0\dots 01\rangle$ component to see if f is constant or balanced.

III. GEOMETRIC ALGEBRA AND ITS BINARY PARAMETRIZATION

Here and in the next section we repeat the presentation from [5].

Euclidean-space GA is constructed as follows. One takes an n-dimensional linear space with orthonormal basis $\{e_1,\dots,e_n\}$. Directed subspaces are then associated with the set

$$\{1, e_1, \dots, e_n, e_{12}, e_{13}\dots, e_{n-1,n}, \dots, e_{12\dots n}\}. \tag{23}$$

Here 1 corresponds to scalars, i.e. a 0-dimensional space. Then we have vectors (oriented segments), bivectors (oriented parallelograms), and so on. There exists a natural parametrization: $1 = e_{0\dots0}$, $e_1 = e_{10\dots0}$, $e_2 = e_{010\dots0}$, $\dots$, $e_{125} = e_{110010\dots0}$, $\dots$, $e_{12\dots n-1,n} = e_{11\dots1}$, which shows that there is a one-to-one relation between an n-bit number and an element of GA. An element with k 1s and $n-k$ 0s is called a k-blade.

A *geometric product* of k 1-blades is a k-blade. For example, $e_{1248} = e_1e_2e_4e_8$. Moreover, $e_ne_m = -e_me_n$, if $m \neq n$, and $e_ne_n = 1$, for any n. GA is a Clifford algebra [12] enriched by certain geometric interpretations and operations.

Particularly interesting is the form of the geometric product that occurs in the binary parametrization. Let us work out a few examples:

$$\begin{aligned}
e_1e_1 &= e_{10\dots0}e_{10\dots0} = 1 = e_{0\dots0} = e_{(10\dots0)\oplus(10\dots0)} &(24)\\
e_1e_{12} &= e_{10\dots0}e_{110\dots0} = e_1e_1e_2 = e_2 = e_{010\dots0} = e_{(10\dots0)\oplus(110\dots0)} &(25)\\
e_{12}e_1 &= e_{110\dots0}e_{10\dots0} = e_1e_2e_1 = -e_2e_1e_1 = -e_2 = -e_{010\dots0} = -e_{(110\dots0)\oplus(10\dots0)} &(26)\\
e_{1257}e_{26} &= e_{11001010\dots0}e_{0100010\dots0} = e_1e_2e_5e_7e_2e_6 = (-1)^2e_1e_2e_2e_5e_7e_6 = (-1)^2(-1)^1e_1e_2e_2e_5e_6e_7\\
&= (-1)^3e_1e_5e_6e_7 = (-1)^3e_{10001110\dots0} = (-1)^De_{(11001010\dots0)\oplus(0100010\dots0)}. &(27)
\end{aligned}$$

The number D is the number of times a 1 from the right string had to "jump" over a 1 from the left one during the process of shifting the right string to the left. Symbolically the operation can be represented as

$$\begin{bmatrix} & \longleftarrow\ 01000100\dots0\\ 11001010\dots0 & \end{bmatrix} \mapsto (-1)^D\begin{bmatrix}01000100\dots0\\11001010\dots0\end{bmatrix} \mapsto (-1)^D\begin{bmatrix}01000100\dots0\\ \oplus \\11001010\dots0\end{bmatrix} = (-1)^D\begin{bmatrix}10001110\dots0\end{bmatrix}$$

The above observations, generalized to arbitrary strings of bits, yield

$$e_{A_1\dots A_n}e_{B_1\dots B_n} = (-1)^{\sum_{k<l}B_kA_l}e_{(A_1\dots A_n)\oplus(B_1\dots B_n)}. \tag{28}$$

Indeed, for two arbitrary strings of bits we have

$$\begin{bmatrix} & \longleftarrow\ B_1B_2\dots B_n\\ A_1A_2\dots A_n & \end{bmatrix} \mapsto (-1)^D\begin{bmatrix}B_1B_2\dots B_n\\A_1A_2\dots A_n\end{bmatrix} \tag{29}$$

where

$$D = B_1(A_2+\dots+A_n) + B_2(A_3+\dots+A_n) + \dots + B_{n-1}A_n = \sum_{k<l}B_kA_l. \tag{30}$$

IV. CARTAN REPRESENTATION

In this section we give an explicit matrix representation of GA. We begin with Pauli's matrices

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{31}$$

GA of a plane is represented as follows: $1 = 2 \times 2$ unit matrix, $e_1 = \sigma_1$, $e_2 = \sigma_2$, $e_{12} = \sigma_1\sigma_2 = i\sigma_3$. Alternatively, we can write $e_{00} = 1$, $e_{10} = \sigma_1$, $e_{01} = \sigma_2$, $e_{11} = i\sigma_3$, and

$$\alpha_{00}e_{00} + \alpha_{10}e_{10} + \alpha_{01}e_{01} + \alpha_{11}e_{11} = \begin{pmatrix} \alpha_{00} + i\alpha_{11} & \alpha_{10} - i\alpha_{01} \\ \alpha_{10} + i\alpha_{01} & \alpha_{00} - i\alpha_{11} \end{pmatrix}. \tag{32}$$

This is equivalent to encoding $2^2 = 4$ real numbers into two complex numbers.

In 3-dimensional space we have $1 = 2 \times 2$ unit matrix, $e_1 = \sigma_1$, $e_2 = \sigma_2$, $e_3 = \sigma_3$, $e_{12} = \sigma_1\sigma_2 = i\sigma_3$, $e_{13} = \sigma_1\sigma_3 = -i\sigma_2$, $e_{23} = \sigma_2\sigma_3 = i\sigma_1$, $e_{123} = \sigma_1\sigma_2\sigma_3 = i$.

Now the representation of

$$\sum_{ABC=0,1} \alpha_{ABC}e_{ABC} = \begin{pmatrix} \alpha_{000} + i\alpha_{111} + \alpha_{001} + i\alpha_{110}, & \alpha_{100} + i\alpha_{011} - i\alpha_{010} - \alpha_{101} \\ \alpha_{100} + i\alpha_{011} + i\alpha_{010} + \alpha_{101}, & \alpha_{000} + i\alpha_{111} - \alpha_{001} - i\alpha_{110} \end{pmatrix} \tag{33}$$

is equivalent to encoding $2^3 = 8$ real numbers into 4 complex numbers.

An arbitrary n-bit record can be encoded into the matrix algebra known as Cartan's representation of Clifford argebras [12]:

$$e_{2k} = \underbrace{\sigma_1 \otimes \cdots \otimes \sigma_1}_{n-k} \otimes \sigma_2 \otimes \underbrace{1 \otimes \cdots \otimes 1}_{k-1}, \tag{34}$$

$$e_{2k-1} = \underbrace{\sigma_1 \otimes \cdots \otimes \sigma_1}_{n-k} \otimes \sigma_3 \otimes \underbrace{1 \otimes \cdots \otimes 1}_{k-1}. \tag{35}$$

V. GA FORMULATION OF THE DEUTSCH-JOZSA PROBLEM

Consider an $(n+1)$-dimensional Euclidean space with orthonormal basis $\{e_1, \ldots e_{n+1}\}$, and its associated GA. The basis vector e_{n+1} in binary parametrization corresponds to $e_{0\ldots01}$. Recall that

$$e_{A_1\ldots A_{n+1}} e_{B_1\ldots B_{n+1}} = (-1)^{\sum_{i<j} B_i A_j} e_{(A_1\ldots A_{n+1})\oplus(B_1\ldots B_{n+1})} \tag{36}$$

and, in particular,

$$e_{A_1\ldots A_{n+1}} e_{0\ldots01} = e_{A_1\ldots A_n, A_{n+1}\oplus 1} \tag{37}$$

$$e_{A_1\ldots A_{n+1}} e_{0\ldots010} = (-1)^{A_{n+1}} e_{A_1\ldots A_{n-1}, A_n\oplus 1, A_{n+1}} \tag{38}$$

Consider

$$\begin{aligned}
E_{n+1} &= \sum_{A_1\dots A_{n+1}=0}^{1} e_{A_1\dots A_{n+1}} && (39)\\
E_{n+1}e_{0\dots 010} &= \sum_{A_1\dots A_{n+1}=0}^{1} e_{A_1\dots A_{n+1}} e_{0\dots 010} && (40)\\
&= \sum_{A_1\dots A_{n+1}=0}^{1} (-1)^{A_{n+1}} e_{A_1\dots A_{n-1},A_n\oplus 1,A_{n+1}} && (41)\\
&= \sum_{A_1\dots A_{n-1}A_{n+1}=0}^{1} (-1)^{A_{n+1}} e_{A_1\dots A_{n-1},0\oplus 1,A_{n+1}} + \sum_{A_1\dots A_{n-1}A_{n+1}=0}^{1} (-1)^{A_{n+1}} e_{A_1\dots A_{n-1},1\oplus 1,A_{n+1}} && (42)\\
&= \sum_{A_1\dots A_{n-1}A_{n+1}=0}^{1} (-1)^{A_{n+1}} e_{A_1\dots A_{n-1},1,A_{n+1}} + \sum_{A_1\dots A_{n-1}A_{n+1}=0}^{1} (-1)^{A_{n+1}} e_{A_1\dots A_{n-1},0,A_{n+1}} && (43)\\
&= \sum_{A_1\dots A_{n+1}=0}^{1} (-1)^{A_{n+1}} e_{A_1\dots A_{n+1}} && (44)
\end{aligned}$$

The influence of E_{n+1} on $e_{0\dots 010}$ is similar to (11):

$$\begin{aligned}
E_{n+1}e_{0\dots 010} &= \sum_{A_1\dots A_{n+1}=0}^{1} (-1)^{A_{n+1}} e_{A_1\dots A_{n+1}} && (45)\\
&= \sum_{A_1\dots A_n=0}^{1} \Big(e_{A_1\dots A_n 0} - e_{A_1\dots A_n 1} \Big) && (46)\\
U_{n+1}|\underbrace{0\dots 0}_{n}1\rangle &= \frac{1}{\sqrt{2^{n+1}}} \sum_{A_1\dots A_{n+1}=0}^{1} (-1)^{A_{n+1}} |A_1\dots A_{n+1}\rangle && (47)\\
&= \frac{1}{\sqrt{2^{n+1}}} \sum_{A_1\dots A_n=0}^{1} \Big(|A_1\dots A_n 0\rangle - |A_1\dots A_n 1\rangle \Big) && (48)
\end{aligned}$$

Now assume there exists an oracle E_f that performs

$$E_f e_{A_1\dots A_n A_{n+1}} = e_{A_1\dots A_n,A_{n+1}\oplus f(A_1\dots A_n)} = e_{A_1\dots A_n,A_{n+1}} e_{0\dots 0,f(A_1\dots A_n)} \qquad (49)$$

Then

$$\begin{aligned}
E_f E_{n+1} e_{0\dots 010} &= \sum_{A_1\dots A_n=0}^{1} E_f \Big(e_{A_1\dots A_n 0} - e_{A_1\dots A_n 1} \Big) && (50)\\
&= \sum_{A_1\dots A_n=0}^{1} E_f \Big(e_{A_1\dots A_n 0} - e_{A_1\dots A_n 1} \Big) && (51)\\
&= \sum_{A_1\dots A_n=0}^{1} \Big(e_{A_1\dots A_n,f(A_1\dots A_n)} - e_{A_1\dots A_n,\neg f(A_1\dots A_n)} \Big) && (52)\\
&= \sum_{A_1\dots A_n=0}^{1} (-1)^{f(A_1\dots A_n)} \Big(e_{A_1\dots A_n 0} - e_{A_1\dots A_n 1} \Big) && (53)
\end{aligned}$$

In GA there exists an operation of *reverse* which reverses the order as follows: If $X = e_1 e_2 \dots e_{k-1} e_k$ then the reverse of X is

$$X^\dagger = e_k e_{k-1} \dots e_2 e_1 = (-1)^{k(k-1)/2} X \qquad (54)$$

By linearity we extend it to all multivectors. In binary parametrization the number k describes the number of 1s in $e_{A_1\ldots A_{n+1}}$, i.e. $k=\sum_{j=1}^{n+1} A_j$. So consider

$$F_{n+1} = \sum_{A_1\ldots A_n=0}^{1} e^{\dagger}_{A_1\ldots A_n 0} \tag{55}$$

$$= \sum_{A_1\ldots A_n=0}^{1} (-1)^{k(k-1)/2} e_{A_1\ldots A_n 0} \tag{56}$$

Here $k=\sum_{j=1}^{n} A_j$ since the last bit is 0. Now

$$F_{n+1}E_f E_{n+1} e_{0\ldots 010} = \sum_{A_1\ldots A_n=0}^{1} (-1)^{f(A_1\ldots A_n)} F_{n+1}\Big(e_{A_1\ldots A_n 0} - e_{A_1\ldots A_n 1}\Big) \tag{57}$$

$$= \sum_{A_1\ldots A_n=0}^{1} (-1)^{f(A_1\ldots A_n)} F_{n+1} e_{A_1\ldots A_n 0} + \ldots \tag{58}$$

$$= \sum_{A_1\ldots A_n=0}^{1} (-1)^{f(A_1\ldots A_n)} \sum_{B_1\ldots B_n=0}^{1} (-1)^{k(k-1)/2} e_{B_1\ldots B_n 0} e_{A_1\ldots A_n 0} + \ldots \tag{59}$$

$$= \sum_{A_1\ldots A_n=0}^{1} (-1)^{f(A_1\ldots A_n)} \sum_{B_1\ldots B_n=0}^{1} (-1)^{k(k-1)/2} (-1)^{\sum_{k<l} A_k B_l} e_{(B_1\ldots B_n 0)\oplus(A_1\ldots A_n 0)} + \ldots \tag{60}$$

$$= \sum_{A_1\ldots A_n=0}^{1} (-1)^{f(A_1\ldots A_n)} e_{0\ldots 0} + \ldots \tag{61}$$

The dots denote all those term where the binary indices contain at least one 1. The two powers of -1 have cancelled out since $e^{\dagger}_{A_1\ldots A_{n+1}} e_{A_1\ldots A_{n+1}} = 1 = e_{0\ldots 0}$. Finally

$$F_{n+1}E_f E_{n+1} e_{0\ldots 010} = \sum_{A_1\ldots A_n=0}^{1} (-1)^{f(A_1\ldots A_n)} e_{0\ldots 0} + \ldots \tag{62}$$

Now let Π project on $1 = e_{0\ldots 0}$. It follows that

$$\operatorname{Tr}\Pi(F_{n+1}E_f E_{n+1} e_{0\ldots 010}) = \sum_{A_1\ldots A_n=0}^{1} (-1)^{f(A_1\ldots A_n)} \operatorname{Tr} e_{0\ldots 0} = N \sum_{A_1\ldots A_n=0}^{1} (-1)^{f(A_1\ldots A_n)} \tag{63}$$

$$= \begin{cases} (-1)^{f(0\ldots 0)} N 2^n & \text{if } f \text{ is constant} \\ 0 & \text{if } f \text{ is balanced} \end{cases} \tag{64}$$

Here $N = \operatorname{Tr} 1$ is the dimension of the representation of GA. We have achieved the same goal as the quantum algorithm.

We have to point out at this moment a possible error one can make. Let us note that in the step

$$E_f e_{A_1\ldots A_n A_{n+1}} = e_{A_1\ldots A_n, A_{n+1}} e_{0\ldots 0, f(A_1\ldots A_n)} \tag{65}$$

we have E_f on the left and $e_{0\ldots 0, f(A_1\ldots A_n)}$ on the right. It might appear that it would be simpler and more natural to write E_f on the right as well. However, this would be misleading since

$$F_{n+1}E_f e_{A_1\ldots A_n A_{n+1}} = F_{n+1}\Big(e_{A_1\ldots A_n A_{n+1}} e_{0\ldots 0, f(A_1\ldots A_n)}\Big) \tag{66}$$

$$\neq \Big(F_{n+1} e_{A_1\ldots A_n A_{n+1}}\Big) e_{0\ldots 0, f(A_1\ldots A_n)} = E_f F_{n+1} e_{A_1\ldots A_n A_{n+1}} \tag{67}$$

VI. EXPLICIT EXAMPLES

A. Two bits

GA of a plane consists of: $1 = 2 \times 2$ unit matrix, $e_1 = \sigma_1$, $e_2 = \sigma_2$, $e_{12} = \sigma_1\sigma_2 = i\sigma_3$. Alternatively, we can write $e_{00} = 1$, $e_{10} = \sigma_1$, $e_{01} = \sigma_2$, $e_{11} = i\sigma_3$.

$$\alpha_{00}e_{00} + \alpha_{10}e_{10} + \alpha_{01}e_{01} + \alpha_{11}e_{11} = \begin{pmatrix} \alpha_{00} + i\alpha_{11} & \alpha_{10} - i\alpha_{01} \\ \alpha_{10} + i\alpha_{01} & \alpha_{00} - i\alpha_{11} \end{pmatrix}. \tag{68}$$

$$E_2 = \begin{pmatrix} 1+i & 1-i \\ 1+i & 1-i \end{pmatrix}. \tag{69}$$

$$F_2 \;=\; e_{00}^\dagger + e_{10}^\dagger = e_{00} + e_{10} = 1 + \sigma_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \tag{70}$$

$$\begin{aligned} E_2 e_{10} &= \Big(e_{00} + e_{10} + e_{01} + e_{11}\Big)e_{10} = e_{10} + e_{00} - e_{11} - e_{01} && (71)\\ &= \Big(1 + \sigma_1 + \sigma_2 + i\sigma_3\Big)\sigma_1 = \sigma_1 + 1 - i\sigma_3 - \sigma_2 && (72)\end{aligned}$$

$$\begin{aligned} E_f E_2 e_{10} &= E_f\Big(e_{10} + e_{00} - e_{11} - e_{01}\Big) && (73)\\ &= e_{1,0\oplus f(1)} + e_{0,0\oplus f(0)} - e_{1,1\oplus f(1)} - e_{0,1\oplus f(0)} && (74)\\ &= e_{1,f(1)} + e_{0,f(0)} - e_{1,\neg f(1)} - e_{0,\neg f(0)} && (75)\end{aligned}$$

1. *Case* $f(0) = f(1) = 0$

$$\begin{aligned} E_f E_2 e_{10} &= e_{1,f(1)} + e_{0,f(0)} - e_{1,\neg f(1)} - e_{0,\neg f(0)} && (76)\\ &= e_{10} + e_{00} - e_{11} - e_{01} && (77)\\ &= \sigma_1 + 1 - i\sigma_3 - \sigma_2 && (78)\\ &= \begin{pmatrix} 1-i & 1+i \\ 1-i & 1+i \end{pmatrix} && (79)\end{aligned}$$

$$F_2 E_f E_2 e_{10} \;=\; \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1-i & 1+i \\ 1-i & 1+i \end{pmatrix} = 2\begin{pmatrix} 1-i & 1+i \\ 1-i & 1+i \end{pmatrix} \tag{80}$$

$$\operatorname{Tr}\Pi F_2 E_f E_2 e_{10} = 4 \tag{81}$$

2. *Case* $f(0) = f(1) = 1$

$$\begin{aligned} E_f E_2 e_{10} &= e_{1,f(1)} + e_{0,f(0)} - e_{1,\neg f(1)} - e_{0,\neg f(0)} && (82)\\ &= e_{11} + e_{01} - e_{10} - e_{00} && (83)\end{aligned}$$

Since this is minus the result from the previus subsection, we immediately get

$$\operatorname{Tr}\Pi F_2 E_f E_2 e_{10} = -4 \tag{84}$$

3. Case $f(0) = 0,\ f(1) = 1$

$$\begin{aligned} E_f E_2 e_{10} &= e_{1,f(1)} + e_{0,f(0)} - e_{1,\neg f(1)} - e_{0,\neg f(0)} & (85)\\ &= e_{11} + e_{00} - e_{10} - e_{01} & (86)\\ &= i\sigma_3 + 1 - \sigma_1 - \sigma_2 & (87)\\ &= \begin{pmatrix} 1+i & -1+i \\ -1-i & 1-i \end{pmatrix} & (88)\end{aligned}$$

$$F_2 E_f E_2 e_{10} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1+i & -1+i \\ -1-i & 1-i \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (89)$$

$$\operatorname{Tr} \Pi F_2 E_f E_2 e_{10} = 0 \quad (90)$$

Alternatively

$$\begin{aligned} F_2 E_f E_2 e_{10} &= (1+\sigma_1)(i\sigma_3 + 1 - \sigma_1 - \sigma_2) & (91)\\ &= (i\sigma_3 + 1 - \sigma_1 - \sigma_2) + (i\sigma_1\sigma_3 + \sigma_1 - 1 - \sigma_1\sigma_2) & (92)\\ &= (i\sigma_3 + 1 - \sigma_1 - \sigma_2) + (i(-i\sigma_2) + \sigma_1 - 1 - i\sigma_3) = 0 & (93)\end{aligned}$$

4. Case $f(0) = 1,\ f(1) = 0$

$$\begin{aligned} E_f E_2 e_{10} &= e_{1,f(1)} + e_{0,f(0)} - e_{1,\neg f(1)} - e_{0,\neg f(0)} & (94)\\ &= e_{10} + e_{01} - e_{11} - e_{00} & (95)\end{aligned}$$

This is minus the result from the previou section and therefore $\operatorname{Tr} \Pi F_2 E_f E_2 e_{10} = 0$.

Summing up, constant functions were producing $2(-1)^{f(0)}2^1$, and balanced functions implied 0, as it should be on general grounds.

B. Three bits

In 3-dimensional space we have $1 = 2\times 2$ unit matrix, $e_1 = \sigma_1$, $e_2 = \sigma_2$, $e_3 = \sigma_3$, $e_{12} = \sigma_1\sigma_2 = i\sigma_3$, $e_{13} = \sigma_1\sigma_3 = -i\sigma_2$, $e_{23} = \sigma_2\sigma_3 = i\sigma_1$, $e_{123} = \sigma_1\sigma_2\sigma_3 = i$. The operation $\operatorname{Tr}\Pi$ corresponds in this representation to taking the real part of trace (only $e_{000} = 1$ and $e_{111} = i$ have nonzero trace).

Now the representation of a general element reads

$$\sum_{ABC=0,1} \alpha_{ABC} e_{ABC} = \begin{pmatrix} \alpha_{000} + i\alpha_{111} + \alpha_{001} + i\alpha_{110}, & \alpha_{100} + i\alpha_{011} - i\alpha_{010} - \alpha_{101} \\ \alpha_{100} + i\alpha_{011} + i\alpha_{010} + \alpha_{101}, & \alpha_{000} + i\alpha_{111} - \alpha_{001} - i\alpha_{110} \end{pmatrix} \quad (96)$$

$$E_3 = \sum_{ABC=0,1} e_{ABC} = \begin{pmatrix} 1+i+1+i, & 1+i-i-1 \\ 1+i+i+1, & 1+i-1-i \end{pmatrix} = 2\begin{pmatrix} 1+i, & 0 \\ 1+i, & 0 \end{pmatrix} \quad (97)$$

$$\begin{aligned} F_3 &= \sum_{AB=0,1} e^\dagger_{AB0} = e^\dagger_{000} + e^\dagger_{100} + e^\dagger_{010} + e^\dagger_{110} = 1 + \sigma_1 + \sigma_2 + (\sigma_1\sigma_2)^\dagger & (98)\\ &= 1 + \sigma_1 + \sigma_2 + \sigma_2\sigma_1 = 1 + \sigma_1 + \sigma_2 - i\sigma_3 = \begin{pmatrix} 1-i & 1-i \\ 1+i & 1+i \end{pmatrix} & (99)\end{aligned}$$

$$
\begin{aligned}
E_3 e_{010} &= 2\begin{pmatrix} 1+i & 0 \\ 1+i & 0 \end{pmatrix}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = 2\begin{pmatrix} 0 & 1-i \\ 0 & 1-i \end{pmatrix} && (100)\\
&= \Big(e_{000} + e_{100} + e_{010} + e_{001} + e_{110} + e_{011} + e_{101} + e_{111}\Big)e_{010} && (101)\\
&= e_{000}e_{010} + e_{100}e_{010} + e_{010}e_{010} + e_{001}e_{010} + e_{110}e_{010} + e_{011}e_{010} + e_{101}e_{010} + e_{111}e_{010} && (102)\\
&= e_{010} + e_{110} + e_{000} - e_{011} + e_{100} - e_{001} - e_{111} - e_{101} && (103)\\
&= \sigma_2 + \sigma_1\sigma_2 + 1 - \sigma_2\sigma_3 + \sigma_1 - \sigma_3 - \sigma_1\sigma_2\sigma_3 - \sigma_1\sigma_3 && (104)\\
&= \sigma_2 + i\sigma_3 + 1 - i\sigma_1 + \sigma_1 - \sigma_3 - i + i\sigma_2 && (105)\\
&= \begin{pmatrix} i+1-1-i, & -i-i+1+1 \\ i-i+1-1, & -i+1+1-i \end{pmatrix} && (106)
\end{aligned}
$$

$$
\begin{aligned}
E_f E_3 e_{010} &= e_{01,0\oplus f(01)} + e_{11,0\oplus f(11)} + e_{00,0\oplus f(00)} - e_{01,1\oplus f(01)} + e_{10,0\oplus f(10)} - e_{00,1\oplus f(00)} - e_{11,1\oplus f(11)} - e_{10,1\oplus f(10)} \\
&= e_{01,f(01)} + e_{11,f(11)} + e_{00,f(00)} - e_{01,\neg f(01)} + e_{10,f(10)} - e_{00,\neg f(00)} - e_{11,\neg f(11)} - e_{10,\neg f(10)} \\
&= e_{00,f(00)} - e_{00,\neg f(00)} + e_{01,f(01)} - e_{01,\neg f(01)} + e_{10,f(10)} - e_{10,\neg f(10)} + e_{11,f(11)} - e_{11,\neg f(11)} && (107)
\end{aligned}
$$

1. Case of constant f, $f(00) = 0$

$$
\begin{aligned}
E_f E_3 e_{010} &= e_{000} - e_{001} + e_{010} - e_{011} + e_{100} - e_{101} + e_{110} - e_{111} && (108)\\
&= 1 - \sigma_3 + \sigma_2 - i\sigma_1 + \sigma_1 + i\sigma_2 + i\sigma_3 - i && (109)\\
&= \begin{pmatrix} 1-1+i-i, & -i-i+1+1 \\ i-i+1-1, & 1+1-i-i \end{pmatrix} = 2(1-i)\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} && (110)
\end{aligned}
$$

$$
\begin{aligned}
F_3 E_f E_3 e_{010} &= \begin{pmatrix} 1-i & 1-i \\ 1+i & 1+i \end{pmatrix} 2(1-i)\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} && (111)\\
&= 2(1-i)\begin{pmatrix} 1-i & 1-i \\ 1+i & 1+i \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} && (112)\\
&= 2(1-i)\begin{pmatrix} 0 & 2(1-i) \\ 0 & 2(1+i) \end{pmatrix} && (113)
\end{aligned}
$$

$$
\Re\,\mathrm{Tr}\,F_3 E_f E_3 e_{010} = 2(1-i)2(1+i) = 8 = 2(-1)^0 2^2 \qquad (114)
$$

2. Case of constant f, $f(00) = 1$

$$
E_f E_3 e_{010} = e_{001} - e_{000} + e_{011} - e_{010} + e_{101} - e_{100} + e_{111} - e_{110} \qquad (115)
$$

$$
\Re\,\mathrm{Tr}\,F_3 E_f E_3 e_{010} = -2(1-i)2(1+i) = -8 = 2(-1)^1 2^2 \qquad (116)
$$

3. Case of balanced f, $f(00) = 0$, $f(10) = 0$

$$
\begin{aligned}
E_f E_3 e_{010} &= e_{00,f(00)} + e_{10,f(10)} + e_{01,f(01)} + e_{11,f(11)} - e_{00,\neg f(00)} - e_{10,\neg f(10)} - e_{01,\neg f(01)} - e_{11,\neg f(11)} \\
&= e_{000} + e_{100} + e_{011} + e_{111} - e_{001} - e_{101} - e_{010} - e_{110} \\
&= e_{000} + e_{100} - e_{010} - e_{001} + e_{011} - e_{101} - e_{110} + e_{111} \\
&= \begin{pmatrix} 1+i-1-i, & 1+i+i+1 \\ 1+i-i-1, & 1+i+1+i \end{pmatrix} = 2(1+i)\begin{pmatrix} 0, & 1 \\ 0, & 1 \end{pmatrix} && (117)
\end{aligned}
$$

$$F_3 E_f E_3 e_{010} \;=\; 2(1+i)\begin{pmatrix} 0 & 2(1-i) \\ 0 & 2(1+i) \end{pmatrix} \tag{118}$$

$$\Re\,\mathrm{Tr}\,F_3 E_f E_3 e_{010} \;=\; \Re 2(1+i)2(1+i) = \Re 8i = 0 \tag{119}$$

VII. FINAL REMARKS

The above examples show that GA allows for a host of new mathematical tricks with respect to standard quantum computation. The representations of binary numbers are different. There is no distinction between "state vectors" and "operators". One can multiply "state vectors" without increasing the dimension. In the above examples both 2-bit and 3-bit problems were represented by 2×2 matrices, a fact showing that one may expect GA to involve less redundancy than standard tensor representations. One can speak of entanglement in GA representations even though the "states" are not tensored with one another. Here again one finds close analogies to what is known from HRRs and BSC. And, last but not least, it seems there is no general difficulty with translating quantum operations into GA forms, and one can expect all quantum algorithms to have GA analogues.

[1] P. Smolensky, Tensor product variable binding and the representation of symbolic structures in connectionist systems, Artificial Intelligence **46**, 159-216 (1990).

[2] P. Kanerva, Binary spatter codes of ordered k-tuples, *Artificial Neural Networks–ICANN Proceedings*, Lecture Notes in Computer Science vol. 1112, pp. 869-873, C. von der Malsburg *et al.* (Eds.) (Springer, Berlin, 1996).

[3] T. Plate, Holographic reduced representations, IEEE Transactions on Neural Networks **6**, 623-641 (1995).

[4] D. Aerts and M. Czachor, Quantum aspects of semantic analysis and symbolic artificial intelligence, Journal of Physics A **37**, L123-L132 (2004).

[5] D. Aerts, M. Czachor, and B. De Moor, On Geometric Algebra representations of Binary Spatter Codes, cs.AI/0610075 (2006).

[6] D. Hestenes and G. Sobczyk, *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics* (Reidel, Dordrecht, 1984).

[7] G. Sommer (ed.), *Geometric Computing with Clifford Algebras* (Springer, Berlin 2001).

[8] D. Deutsch and R. Jozsa, Rapid solutions of problems by quantum computation, Proceedings of the Royal Society of London A **439**, 553-558 (1992).

[9] S. Somaroo, D. G. Cory, and T. F. Havel, Expressing the operations of quantum computing in multiparticle geometric algebra, Physics Letters A **240**, 1-7 (1998).

[10] T. F. Havel and C. J. L. Doran, Geometric algebra in quantum information processing, quant-ph/0004031.

[11] M. Van den Nest, J. Dehaene, and B. De Moor, Finite set of invariants to characterize local Clifford equivalence of stabilizer states, Physical Review A **72**, 014317 (2005).

[12] P. Budinich and A. Trautman, *The Spinorial Chessboard* (Springer, Berlin, 1988).